Membership Query Removal Implies Non-trivial Cryptographic Primitives

John Wright

May 17, 2010

1 Introduction

1.1 Overview

Understanding learning and discovering to what extent it can be automated is one of the major problems in computer science. This is of obvious relevance when trying to create intelligent robots, for example, and fields as diverse as biology and economics use learning algorithms to help manage and understand their extremely large data sets. There has even been work devoted to analyzing computational learning systems in order to gain intuition about natural learning systems: authors of a '60s textbook on learning machines called perceptrons give voice to their hope that among their readership are "psychologists and biologists who would like to know how the brain computes thoughts" [MP69]. More recently, we have seen the development of a mathematical model for analyzing evolution as a learning process [Val09].

One major approach taken to analyzing learning has been to look at classification problems. As an example of such a problem, consider the scenario of teaching a computer to tell when an image has a bird in it. Say you go about doing this by showing the computer a ton of pictures and telling it which ones have birds in them. After giving it some time to compute, you might hope that it can now figure out whether or not images that it hasn't seen before have birds in them. But how can you be sure that this is a fruitful way of teaching image recognition?

Computational learning theory attempts to answer this question in a mathematically rigorous way by formalizing these classification scenarios in models that define exactly the interaction between the learner and what it's trying to learn and give a metric for evaluating the learner's success. When learning is so formalized in models, we can begin to answer natural questions about learning, such as to what extent does asking questions help. Two of these models that we will focus on here are Leslie Valiant's Probably Approximately Correct (PAC) model of learning and Dana Angluin's exact learning model.

Researchers have carefully studied the issue of whether giving the learner the ability to ask questions (or to pose membership queries, in learning theory terminology) improves its ability to learn, and the answer is dependent on the learning model under consideration. In the agnostic learning model, for instance, a learner which works with membership queries can be converted to one which

works without membership queries, although in the distribution-specific agnostic setting membership queries do increase the power of the learner [Fel09]. In both the PAC model and the exact learning model, on the other hand, it has been shown that there are some concepts that can be learned with membership queries but cannot be learned without. A well known example of such a class is the class of deterministic finite automatas (DFAs), which were shown to be learnable with membership queries in [Ang87] but not learnable without membership queries in [KV94] given suitable assumptions.

Another major development of computer science has been that of modern cryptography. Starting from certain basic and currently unavoidable assumptions, powerful cryptographic protocols have been developed which multiple consenting parties can use to engage in secure communication, which has a wide variety of uses, including electronic commerce. A major part of the cryptographic research effort has been to further understand the assumptions that form the foundation of these cryptographic protocols, with one eventual aim being perhaps to construct cryptographic protocols that do not require any assumptions. An example of a protocol that will be used in this paper is the signature scheme, which allows a user to sign any message they send so that whoever receives it knows who sent it.

The link between cryptography and learning theory was recognized very early on. The very first paper on PAC learning rules out learning polynomial-size circuits given cryptographic assumptions [Val84]. At a high level, a cryptographic protocol proven to be secure has a guarantee attached to it which prevents a malicious user from breaking the security, no matter how well that user may learn. So provably-secure cryptography bounds the ability to learn.

One such link was shown by Angluin and Kharitonov in [AK95]. They showed that giving learners membership queries in the PAC model does not help them for a large number of concepts being learned if signature schemes exist. In this paper, we investigate whether the converse of this statement can be proven, i.e. whether it can be shown that if making membership queries does not help learners, then signature schemes exist. We look at severely weakened versions of the converse and prove a couple of results. On the whole, however, our attempt at converting their result is largely unsuccessful.

1.2 Outline

First, we will describe in detail the result of Angluin and Kharitonov. Their result provides much of the foundation of the approach we take in this paper, and the intuition used in proving their result is needed in proving our results as well. Some of the background and definitions used in this section are also used in later sections.

Second, we give background material for the next section. Our results in the next section will take the form (assumption about query removing) implies (cryptographic primitive with security property), and it is in this section that we lay out all the assumptions and security properties and compare them.

Third, we show our two original results along with one approach that did not work. The two results are two different ways of weaking the converse into a provable statement. The one failed approach is an attempt at a result stronger than the first two, and we explore exactly why it fails.

Finally, we take a look at what has been accomplished and try to judge to what extent it has been a success.

2 The Angluin and Kharitonov Result

In this section we give an overview of the Angluin and Kharitonov result. There are some complicating details that are omitted, and the proof sketch at the end is meant to provide motivation and intuition for our results and approaches that follow rather than to be a completely rigorous statement of the proof provided in their paper. As a result, we slightly fudge some of the numbers for increased clarity.

We make use of the following definition throughout this paper: a function $f(\cdot)$ is **negligible** in n if for all polynomials $p(\cdot)$, there exists an $N > 0$ such that for all $n > N$, $f(n) < \frac{1}{p(n)}$. In other words, f is very small.

2.1 PAC Learning

The Angluin and Kharitonov result is a statement about learning in Leslie Valiant's Probably Approximately Correct (PAC) model. In broad terms, this model places the learner in a classification scenario, meaning it must classify Boolean strings based on whether they have some property. This property is represented as a concept $c \subseteq \{0,1\}^n$, which is identified with the Boolean indicator function $c : \{0,1\}^n \to \{0,1\}$, where $c(x) = 1 \iff x \in c$. Letting C_n be a set of concepts on n variables (i.e. $c \in C_n$), we define a concept class to be the set of all concepts $C = \bigcup_{n \geq 1} C_n$. Furthermore, there is some reasonable encoding of the concepts which gives a measurement $size(c)$ of how complex the concepts are. For example, the size of a $3CNF$ is the number of clauses it contains. It is reasonable to allow the learner more time to run when learning more complex concepts.

The learning scenario is as follows: the learner L wants to learn some $c : \{0,1\}^n \to \{0,1\}$. There is some underlying probability distribution on Boolean strings D. The learner's access to c is mediated through an example oracle $EX(c, D)$, which when called will return to L a tuple $(x, c(x))$, where x is drawn from D. The learner may run in polynomial time, and when it has finished it outputs a hypothesis $h : \{0,1\}^n \to \{0,1\}$. The error of this hypothesis is $err(h) = Pr_{x \sim D}[c(x) \neq h(x)]$. There are two error inputs $0 < \epsilon, \delta, < 1$, and the expectation is that $err(h) \leq \epsilon$ with success probability at least $1 - \delta$. Thus,

Definition 1. *L PAC learns C if for any $n \geq 1$, distribution D on n variables, concept $c \in C_n$, and $0 < \epsilon, \delta, < 1$, L, when given access to $EX(c, D)$, outputs in time $poly(n, size(c), \epsilon^{-1}, \delta^{-1})$ a hypothesis $h : \{0,1\}^n \to \{0,1\}$ such that with probability at least $1 - \delta$, $err(h) \leq \epsilon$.*

Additionally, we can consider the option of giving L the ability to make membership queries, in which it is able to pose examples of its own as well as receiving them from the example oracle. Thus, if L makes a membership query on x, it will receive $c(x)$. If L learns C with membership queries, we say that L PAC learns C with membership queries.

2.2 Signature Schemes

Their result also makes use of cryptographic protocols called signature schemes. A signature scheme is a tuple of algorithms $SIGSCHEME = (GEN, SIGN, VER)$ which obey certain properties. GEN is a randomized polynomial-time algorithm which takes as input n in unary and outputs (PK, SK), the public and secret keys. $SIGN$ is a randomized polynomial-time algorithm which takes as input the two keys and the message m to be signed and outputs a signature $s(m)$ of that message of length $p(n)$, for some polynomial $p(\cdot)$. VER is a deterministic polynomial-time algorithm which takes as input the public key, a message m, and a potential signature $s?(m)$, and verifies whether $s?(m)$ is a proper signature of m. VER is required to give a positive verification for any signature produced for a message by $SIGN$.

The signature scheme is required to satisfy certain properties to be considered secure, and these properties differ based on the level and type of security expected of the signature scheme. Angluin and Kharitonov require that their signature schemes be secure against existential forgery. The scenario is as follows: the forger F is given access to both the public key PK and a signing oracle $SIGN\_ORACLE(\cdot)$, which when called on m will return a proper signature of m, i.e. $SIGN(PK, SK, m)$. In other words, F is granted black box access to the signing algorithm, in addition to the public information it already has. When F is done running, it is expected to produce a tuple $(m, s?(m))$, where m is a never-before-seen message and $s?(m)$ is an attempted signature forgery. The expectation is that if the signature scheme is secure at all, the probability that F outputs a successful forgery is small. In other words,

Definition 2. *$SIGSCHEME$ is secure against existential forgery if for every forger F and for sufficiently large n, when (PK, SK) is drawn from $GEN(1^n)$ and F is given access to PK and $SIGN\_ORACLE(\cdot)$, the probability that F outputs a tuple $(m, s?(m))$, for m a never-before-seen message, such that $VER(PK, m, s?(m)) = 1$, is negligible in n.*

2.3 Their Result

Their result works by taking as input a learning algorithm which uses membership queries and, through the clever use of a signature scheme, producing a learning algorithm which works without using membership queries. Obviously, this can't work for all learning algorithms, or membership queries would add no power to learners whatsoever. The key is that their membership query remover works by embedding the signature scheme into the concept being learned. Thus, this will not work for certain weaker concept classes which require an exponential increase in the size of their concepts to embed the signature scheme. However, Angluin and Kharitonov stilla attempt to accomodate as many concept classes as possible, and to do this they use of a peculiar object called a *tableau*. A tableau is a complete transcript of the history of a computation on some input. Being given access to tableaus can often simplify computations, thus allowing more concept classes to participate in the query removal. Because VER is a deterministic polynomial time algorithm, its tableaus are also of polynomial length. So the size of the tableau is $t(n)$, for some polynomial $t(\cdot)$. Let $TAB(PK, x, y)$ be the proper tableau for the verification algorithm VER on input (PK, x, y).

The embedding works as follows: for any concept $c \in C$ which takes as input strings $x \in \{0,1\}^n$, there is a corresponding concept c' also in C which takes as input tuples (x, y, z) from a larger input space. For c' to be a proper embedding of the signature scheme into c, any (x, y, z) for which $c'(x, y, z) = 1$ must satisfy certain properties. First, c' must correspond to c, so $c(x) = 1$. Second,

y must be a proper signature of x. Third, z must be a proper tableau for the verification algorithm VER on (x, y). Thus, c' combines both c and the signature scheme, and has some extra information which helps to compute it. What we're interested in is concept classes for which $size(c')$ is not too much larger than $size(c)$. This notion is captured in the following definition:

Definition 3. *A concept class C is suitable for signature schemes if for every signature scheme $SIGSCHEME$ there is some polynomial $q(\cdot)$ such that for every integer n, for every $c \in C$ over n variables, and for every pair of keys (PK, SK) that GEN can output on the input 1^n, there is a $c' \in C$ such that $size(c') \leq q(n) * size(c)$ and $c'(x, y, z) = 1 \iff c(x) = 1$, $VER(PK, x, y) = 1$, and $z = TAB(PK, x, y)$.*

An example of a concept class that us suitable for signature schemes is $3CNF$. Angluin and Kharitonov give a more general version of this embedding which works for a wider collection of concept classes, but the embedding we give here is the foundation of theirs and is more appropriate for what we are trying to accomplish in this section, i.e. to give intuition.

We now sketch out their result:

Theorem 1 (Angluin and Kharitonov)**.** *If signature schemes exist, any concept class suitable for signature schemes is PAC learnable without membership queries if it is PAC learnable with membership queries.*

Proof (Sketch). Let C be a concept class suitable for signature schemes and let QL be an algorithm which PAC learns it with membership queries. We will provide an algorithm that L which PAC learns C without membership queries.

algorithm L = On input n, $size(c)$, ϵ, and δ:

1. $GEN(1^n) \to (PK, SK)$
2. Run $QL(n + p(n) + t(n), q(n) * size(c), \epsilon, \delta)$. When it
3. (a) requests an example, draw an example (x, b) and give QL the tuple (x, y, z, b), where $y = SIGN(PK, SK, x)$ and $z = TAB(PK, x, y)$. Save the tuple (x, b)
   (b) makes a membership query on (x, y, z):
      i. if y is a signature for x, z is the tableau of VER on x and y, and an example of the form (x, b) has been drawn before, answer b.
      ii. otherwise, answer 0.
   (c) outputs a hypothesis h', output h, where $h(x) = h'(x, y, z)$ for $y = SIGN(PK, SK, x))$ and $z = TAB(PK, x, y)$.

Obviously, L runs without membership queries. What we must check now is that it preserves the success characteristics of QL. Assume for a second that every membership query made by QL is answered correctly. Then if c is the concept L is trying to learn, QL is being trained on the concept $c' : \{0,1\}^{n+p(n)+t(n)} \to \{0,1\}$, where $c'(x, y, z) = 1 \iff c(x) = 1$, $Pr[SIGN(PK, SK, x) = y)] \neq 0$, and $z = TAB(PK, x, y)$. Furthermore, if D is the distribution L is trying to learn on, then QL is being trained on the distribution D', where $D'(x, y, z) = D(x) \cdot Pr[SIGN(PK, SK, x) = y)]$ if

$z = TAB(PK, x, y)$ and 0 otherwise. By the guarantee that QL is a valid learning algorithm, under the assumption that its membership queries are answered correctly, the hypothesis h' it produces learns concept c' on distribution D' to an error of $err(h') < \epsilon$ with probability at least $1 - \delta$. This gives us the following bound on the error of L with probability at least $1 - \delta$ (where used, z' will be an appropriate tableau):

$$\begin{aligned} err(h) &= Pr_{x \sim D}[c(x) \neq h(x)] \\ &= Pr_{x \sim D}[c(x) \neq h'(x, SIGN(PK, SK, x), z')] \\ &= Pr_{(x,y,z) \sim D'}[c(x) \neq h'(x, SIGN(PK, SK, x)), z'] \\ &= Pr_{(x,y,z) \sim D'}[c'(x, y, z) \neq h'(x, SIGN(PK, SK, x)), z'] \\ &= Pr_{(x,y,z) \sim D'}[c'(x, y, z) \neq h'(x, y, z)] \\ &= err(h') \\ &\leq \epsilon \end{aligned}$$

Now, the only thing remaining to be accounted for is the possibility that a membership query is answered incorrectly. The only time in which this takes place is when QL queries an (x, y, z) such that $c(x) = 1$, y is a proper signature for x, and z is the tableau of VER on x and y, but no example of the form (x, b) has ever been seen before. In this case, QL has produced, for an x it hasn't yet seen, a signature y such that $VER(PK, x, y) = 1$. Since the signature scheme is secure against existential forgery, the probability that this can occur is negligible in n, which does not add significantly to the probability of failure, and thus we are done.

Angluin and Kharitonov give a more rigorous proof of this by proving the contrapositive, i.e. by showing that if L did not learn properly without membership queries then it could be modified into a forger which breaks the signature scheme's security property. □

This hints at the kind of approach we are going to take in proving our converses: their algorithm has an example map $x \mapsto (x, SIGN(PK, SK, x), z)$. Our approach is to analyze algorithms which use similar example maps and see if these example maps are necessarily some sort of cryptographic primitive.

3 Preliminaries

Ideally, our result would be a complete converse of the Angluin and Kharitonov result, in other words a result stating that if membership queries don't matter for concepts suitable for signature schemes, then signature schemes exist. What we have instead are results that are weaker forms of this statement. All of our results begin by assuming the existence of a process which takes a learner that uses membership queries and then produces one that doesn't. Furthermore, we assume that this process looks like the one used in the Angluin and Kharitonov paper. More formally, we posit the existence of a **query remover** $QREM = (GEN, EM)$ which obeys certain properties. GEN is a randomized polynomial-time algorithm which takes as input n in unary and outputs a key. The example map EM is a polynomial-time algorithm which takes a key and a string of length n and outputs another string of length $p(n)$, for $p(\cdot)$ some polynomial.

We want to use our query remover to help learn concept classes, but, as above, not all concept classes are powerful enough to support query removers. A concept class C is **suitable** for $QREM$ if there

is some polynomial $q(\cdot)$ such that for every integer n, for every $c \in C$ over n variables, and for every key k that GEN can output on the input 1^n, there is a $c' \in C$ such that $size(c') \leq q(n) * size(c)$ and $c'(x, y) = 1 \iff c(x) = 1$ and $EM_k(x) = y$. For the sake of clarity, we have suppressed references to tableaus in this and the following sections, but they could be restored with little change to what follows.

Now we can define the query removal properties that we expect $QREM$ to possess. Let C be a concept class suitable for $QREM$, let QL be a membership query algorithm, and let $p(\cdot)$ and $q(\cdot)$ be the polynomials described above. Consider the following algorithm $L_{QREM}(QL)$:

algorithm $L_{QREM}(QL)$ = On input n, $size(c)$, ϵ, and δ:

1. $GEN(1^n) \to k$
2. Run $QL(n + p(n), q(n) * size(c), \frac{\epsilon}{2}, \frac{\delta}{2})$. When it
3. (a) requests an example, draw an example (x, b) and give QL $((x, EM_k(x)), b)$.
   (b) makes a query for an (x, s) which has not been seen before as an example request, answer 0.
   (c) makes a query for an (x, s) which has been seen before as an example request, answer what was answered before.
   (d) outputs a hypothesis h, output h', where $h'(x) = h(x, EM_k(x))$.

If $QREM$ is to earn its name as a query remover, the transformation from QL to $L_{QREM}(QL)$ must not only get rid of queries (which it obviously does), it must also preserve the success characteristics of QL. In other words, we ask that the following be satisfied.

Assumption 1. *If QL PAC-learns C with membership queries, then $L_{QREM}(QL)$ PAC-learns C (without membership queries).*

This is the main assumption we will be dealing with, but sometimes we will make use of the following stronger assumption.

Assumption 2. *If QL PAC-learns C with membership queries, then the probability that $L_{QREM}(QL)$ answers a membership query from QL incorrectly is negligible in n.*

If $QREM$ satisfies Assumption 2 then it also satisfies Assumption 1, because if $L_{QREM}(QL)$ answers queries correctly an overwhelming amount of the time, then QL is almost always simulated properly, so it will almost always output a correct hypothesis. However, what is not clear is whether Assumption 1 implies Assumption 2.

There is one more assumption which will be of use. In this assumption, we will be shifting our attention to Angluin's exact learning model, and so we will need to perform a slight adjustment to $L_{QREM}(QL)$. But first, we will define the model.

In Angluin's exact learning model, the learner has access to equivalence queries EQ instead of the example oracle from the PAC setting. For an equivalence query, the learner submits a hypothesis h, and $EQ(h)$ tests whether $h = c$. If not, the query returns a counterexample x for which $h(x) \neq c(x)$.

The expectation is that the learner in polynomial time arrives at an equivalence query for which its hypothesis is equivalent to c. If this is always the case, then the learner exactly learns C.

So, we now perform the necessary adjustment of $L_{QREM}(QL)$.

algorithm $L'_{QREM}(QL)$ = On input n, $size(c)$, and δ:

1. $GEN(1^n) \rightarrow k$
2. Run $QL(n + p(n), q(n) * size(c), \frac{\delta}{2})$. When it
3. (a) performs an equivalence query on a hypothesis h, perform an equivalence query on hypothesis h', where $h'(x) = h(x, EM_k(x))$. If a counterexample (x, b) is returned, give QL $((x, EM_k(x)), b)$.
   (b) makes a query for an $(x, s(x))$ which has not been seen before as an example request, answer 0.
   (c) makes a query for an $(x, s(x))$ which has been seen before as an example request, answer what was answered before.
   (d) outputs a hypothesis h, output h', where $h'(x) = h(x, EM_k(x))$.

This is identical to $L_{QREM}(QL)$, except it accounts for the fact that in the exact learning model, equivalence queries, not example oracle requests, are made. This is needed for the following assumption.

Assumption 3. *If QL exactly learns C with membership queries, then $L'_{QREM}(QL)$ exactly learns C (without membership queries).*

Now that we have some hardness assumptions in place, it is natural to use these to define cryptographic tools. In this case, we will be transforming query removers into message authentication schemes (MACs). Given a query remover $QREM = (GEN, EM)$, we define its corresponding MAC as:

Definition 4. *$MAC(QREM) = (GEN, SIGN, VER)$ is the MAC whose calls are handled as:*

1. $SIGN_k(x) = EM_k(x)$
2. $VER_k(x, s) = \begin{cases} 1 & \text{if } EM_k(x) = s \\ 0 & o.w. \end{cases}$

Ideally, we would like this $MAC(QREM)$ to satisfy the following security property.

Security Property 1. *For any forger F, for any distribution D over $\{0,1\}^n$, when k is generated by $GEN(1^n)$ and F is fed examples of the form $(x, SIGN_k(x))$, where the x's are drawn from D, the probability that F outputs a new pair $(y, s?(y))$ such that $VER_k(y, s?(y)) = 1$ is negligible in n.*

A second, weaker, notion of security which we will also use is the following.

Security Property 2. *There is no forger F which, when given a polynomial number of distinct $(x, SIGN_k(x))$ pairs, always outputs a new pair $(y, s?(y))$ such that $VER_k(y, s?(y)) = 1$.*

Notice that this security property is entirely distribution-free, which suggests that it will be used in concert with Assumption 3, the exact learning assumption.

4 Results

Here we present two results that give partial converses to the Angluin and Kharitonov result. Then, we give a possible approach for a stronger converse, and explain why it fails.

4.1 The First Proof

Ideally, we would like a proof showing that if $QREM$ satisfies Assumption 1, then $MAC(QREM)$ satisfies Security Property 1. We don't have this, but in lieu of this result we do have the following.

Theorem 2. *If $QREM$ satisfies Assumption 2, then $MAC(QREM)$ satisfies Security Property 1.*

Proof. We will show this by proving the contrapositive. Assume that $MAC(QREM)$ does not satisfy Security Property 1. Then there exists a forger F and a distribution D over $\{0,1\}^n$ such that when k is generated by $GEN(1^n)$ and F is fed examples of the form $(x, SIGN_k(x))$, where the x's are drawn from D, the probability that F outputs a new pair $(y, s?(y))$ such that $VER_k(y, s?(y)) = 1$ is nonnegligible. Now we will use this to construct a query learner which poses membership queries which $L_{QREM}(QL)$ is forced to get wrong.

Let C be a concept class which is PAC-learnable with membership queries, is suitable for $QREM$, and contains the all-1's concept. Let QL be an algorithm which PAC-learns C with membership queries. Consider the following algorithm QL'.

algorithm QL' = On input n, $size(c)$, ϵ, and δ:

1. Run $QL(n, size(c), \epsilon, \delta)$ until it outputs a hypothesis h.
2. Run F concurrently with QL, giving it samples drawn from the example oracle (first, the samples drawn while running QL, and then independently drawn samples if it requests more). When F outputs a new example, do a membership query on it. (Disregard the result.)
3. Output h.

QL' is a valid PAC learner for C with membership queries because it outputs hypotheses in a manner identical to QL. However, it can be used to make $QREM$ violate Assumption 2.

If we use $L_{QREM}(QL')$ to attempt to learn the all-1's concept on the distribution D, then when QL' is run as a subroutine it will in turn run F, providing it with $(x, EM_k(x))$ pairs, where the x's are drawn from D. With nonnegligible probability F will output a new pair $(y, s?(y))$ in which $VER_k(y, s?(y)) = 1$. By definition, $EM_k(x) = SIGN_k(x)$ and $VER_k(x, s) = 1 \iff EM_k(x) = s$, so $EM_k(y) = s?(y)$. This means that a membership query on $(y, s?(y))$ should return 1, because $L_{QREM}(QL')$ is learning the all-1's function. However, this is a new pair, and $L_{QREM}(QL')$ answers

any membership query it hasn't seen before with a 0. So with nonnegligible probability a membership query is answered incorrectly, violating Assumption 2. □

Theorem 1 is somewhat unsatisfying, as algorithm QL' acts essentially identically regardless of whether it is fed an incorrect query. In other words, it causes the query remover to err, but in a relatively benign way. What we would like is a learning algorithm which, while otherwise behaving well, can detect itself being sent through the query remover, in which case it acts eccentrically. (All of this, of course, assumes we have a working forger.)

4.2 The Second Proof

A possible way to do this is to make a membership query on an example and compare the answer of that query to the answer given by the hypothesis produced by QL on line 1 of the algorithm QL'. Because QL produces highly accurate hypotheses, this membership query test should detect when the query is being answered incorrectly. All that is needed is a membership query which is known to be answered incorrectly, which in Theorem 1 we have seen how to construct. We shift our attention to the exact learning model for this proof.

Theorem 3. *If $QREM$ satisfies Assumption 3, then $MAC(QREM)$ satisfies Security Property 2.*

Proof. We will show this by proving the contrapositive. Assume that $MAC(QREM)$ does not satisfy Security Property 2. Then there exists a polynomial $r(\cdot)$ and a forger F which, when given $r(n)$ distinct $(x, SIGN_k(x))$ pairs, always outputs a new pair $(y, s?(y))$ such that $VER_k(y, s?(y)) = 1$. Now we will use this to construct a query learner which poses membership queries which $L_{QREM}(QL)$ is forced to get wrong.

Let C be a concept class which is exactly learnable with membership queries, is suitable for $QREM$, and contains the all-1's concept. Let QL be an algorithm which exactly learns C with membership queries. Consider the following algorithm QL'.

algorithm QL' = On input n and $size(c)$:

1. Generate $r(n)$ distinct examples using equivalence queries and feed them to F. If it outputs a new example y, perform a membership query on y and save the answer as b.
2. Run $QL(n, size(c))$ until it outputs a hypothesis h.
3. Output the hypothesis h', where $h'(x) = \begin{cases} b & \text{if } x = y \\ h(x) & \text{o.w.} \end{cases}$

QL' is a valid exact learner with membership queries for concept class C, but when attempting to learn the all-1's concept $L'_{QREM}(QL')$ will never work. This is because F is guaranteed to always generate a correct forgery, and when the membership query on this forgery is performed, the response will always be 0. Because we are learning the all-1's concept, it should be 1. This error is carried over into the final hypothesis, and thus the final hypothesis is error-prone. So $QREM$ does not satisfy Assumption 3. □

This gives us a result in the right direction, and poses the question of whether the shift to the exact learning model was necessary.

4.3 The Third Proof

Ideally, we'd like to prove the following statement:

Theorem 4. *(incorrect) If $QREM$ satisfies Assumption 1, then $MAC(QREM)$ satisfies Security Property 1.*

And a proof of this statement might go as follows:

Proof. (incorrect)We will show this by proving the contrapositive. Assume that $MAC(QREM)$ does not satisfy Security Property 1. Then there exists a forger F and a distribution D over $\{0,1\}^n$ such that when k is generated by $GEN(1^n)$ and F is fed examples of the form $(x, SIGN_k(x))$, where the x's are drawn from D, the probability that F outputs a new pair $(y, s?(y))$ such that $VER_k(y, s?(y)) = 1$ is nonnegligible. Now we will use this to construct a query learner which poses membership queries which $L_{QREM}(QL)$ is forced to get wrong.

Let C be a concept class which is PAC-learnable with membership queries, is suitable for $QREM$, and contains the all-1's concept. Let QL be an algorithm which PAC-learns C with membership queries. Consider the following algorithm QL'.

algorithm QL' = On input n, $size(c)$, ϵ, and δ:

1. $min(\epsilon, \delta) \rightarrow \epsilon'$
2. Run $QL(n, size(c), \epsilon', 1 - \frac{1-\delta}{1-\epsilon'})$ until it outputs a hypothesis h.
3. Run F concurrently with QL, giving it samples drawn from the example oracle (first, the samples drawn while running QL, and then independently drawn samples if it requests more). When F outputs a new example $(x, s?(x))$, do a membership query on it. Save the result as bit b.
4. Evaluate $h(x, s?(x))$. If it equals b, output h. Otherwise, output the all-0's hypothesis.

As before, QL' is a valid PAC learner for C with membership queries. This is because with probability $1 - (1 - \frac{1-\delta}{1-\epsilon'}) = \frac{1-\delta}{1-\epsilon'}$, h misclassifies an example drawn from D with probability at most ϵ'. So it will correctly evaluate the forgery produced by F with probability $1 - \epsilon'$, meaning that with probability at least $(1 - (1 - \frac{1-\delta}{1-\epsilon'}))(1 - \epsilon') = \frac{1-\delta}{1-\epsilon'}(1 - \epsilon') = 1 - \delta$, QL' will output an h with error ϵ', which is at most ϵ.

Now, we will show how this makes $QREM$ violate Assumption 1. If we use $L_{QREM}(QL')$ to attempt to learn the all-1's concept on the distribution D, then when QL' is run as a subroutine it will in turn run F, which will produce a valid forgery that should be evaluated to 1. Because h is $1 - \epsilon$ accurate it will with high probability properly evaluate this forgery correctly. However, when the membership query is performed it will almost always be a 0, so QL' will output the all-0's hypothesis. Since

$L_{QREM}(QL')$ is learning the all-1's hypothesis, the hypothesis QL' outputs is 100% wrong, meaning that the query removal process has failed. Thus, $L_{QREM}(QL')$ violates Assumption 1. □

There are many things wrong with this proof, most of which stem from the fact that our forger F does not output forgeries distributed according to D. In fact, it does not even output forgeries which are restricted to the support of D. This is problematic as QL is guaranteed to output hypotheses which are highly accurate only with respect to D. So long as F outputs forgeries from a distribution which is not D, there are no guarantees that h performs well, and thus h is of no use as a means of checking whether the membership queries are being answered correctly. This means QL' is not even necessarily a valid PAC learner for C with membership queries, and so rigging $L_{QREM}(QL')$ to fail gives us nothing.

5 Conclusion

Our first work with the Angluin and Kharitonov paper was to try to see whether any other assumptions (rather than signature schemes) would imply that queries could be removed. This proved to be difficult, however, and it eventually seemed as though signature schemes were exactly what was needed to perform query removal. We took a stab at trying to formalize this, and this paper is the consequence.

Our results are a mixed bag: to get strong security properties we need a very strong assumption. On the other hand, a weakening of the assumption yields a severely crippled cryptographic protocol. And our attempt at finding a happy medium encounters some fundamental problems. On the whole, this appears to be a difficult problem to grapple with.

Membership queries are still providing fruitful avenues of research. The membership query results mentioned in the introduction regarding the agnostic learning model are only two years old, and one of the results left room for improvement. This paper leaves open some substantial room for improvement of its own as well.

Cryptography and learning theory have been joined at the hip for quite some time, and new results linking the two are ever forthcoming. This paper provides yet another link between the two, applying learning theoretic tools to build a foundation for cryptography.

6 Acknowledgments

I would like to thank Professor Adam Klivans and Alex Tang for working with me on this project.

References

[AK95] Dana Angluin and Michael Kharitonov. When won't membership queries help? *J. Comput. Syst. Sci.*, 50(2):336–355, 1995.

[Ang87] Dana Angluin. Learning regular sets from queries and counterexamples. *Inf. Comput.*, 75(2):87–106, 1987.

[Fel09] Vitaly Feldman. On the power of membership queries in agnostic learning. *Journal of Machine Learning Research*, 10:163–182, 2009.

[KV94] Michael J. Kearns and Leslie G. Valiant. Cryptographic limitations on learning boolean formulae and finite automata. *J. ACM*, 41(1):67–95, 1994.

[MP69] Marvin Minsky and Seymour Papert. *Perceptrons: An Introduction to Computational Geometry*. The MIT Press, 1969.

[Val84] Leslie G. Valiant. A theory of the learnable. In *STOC*, pages 436–445. ACM, 1984.

[Val09] Leslie G. Valiant. Evolvability. *J. ACM*, 56(1), 2009.